arXiv:2304.14950v2 [cs.LO] 1 May 2023

动态追踪：用于重写协议的图形语言

Kristopher Brown托扑斯研究所
kris@topos.institute

David I. Spivak托扑斯研究所 david@topos.institute

摘要

范畴 $\mathbf{Set}_*$（由集合和偏函数构成）众所周知是带迹幺半范畴，这意味着偏函数 $S+U \rightharpoonup T+U$ 可以一致地变换为偏函数$S \rightharpoonup T$。这种变换通常通过一个必须执行的隐式过程来描述，

我们通过在 $\mathbf{Cat}^\sharp$ 中丰富这个带迹范畴，使该过程显式化；**Cat** 是由范畴和余函子构成的对称幺半范畴：每个 Hom-范畴都以此类过程为对象，并以过程的推进为态射。我们还将结果推广到 $\mathbf{Set}_*$ 之外的带迹 Kleisli 范畴，为任意形如 $t+1$ 的多项式单子的 Kleisli 范畴提供一个猜想性的迹算子。这项工作的主要动机是为执行由图重写驱动的复杂计算提供一种形式化的图形语法，而图重写本身就是一种用于数据变换的图形语言。

关键词 双推出重写 · 范畴论 · 图重写

1 动机

显式构造的程序是指定数据变换的标准方式。然而，如果不追求通用编程语言的全部表达能力，就可以采用一种受限语法——领域专用语言——并由此获得一些理想的特性。例如，流程图常被用作软件项目的非正式语法，其中每个方框都与一个子程序相关联，该子程序将输入连线上的输入数据转换为输出连线上的输出数据。范畴论使这种语义赋予过程变得精确：有向连线图的语法可以在任意对称幺半范畴中被形式化地赋予语义。与通用语言中的一维表达式树相比，这种二维语法具有诸多优势，包括直观的可视化表示、运算子代入以及代数运算。此外，当图被表示为组合对象时，图本身便成为一种特别高效的范式形式，可对应于大量与之等价的语法树 [18]。

图变换领域采用黏着范畴中跨度的语法，并将其解释为重写规则。在各种形式体系中，尤其是 DPO [7], SPO [13], 和 SqPO [6] 重写中，删除、复制、合并和添加数据的语义都可以归因于这些跨度。如果所需的操作仅限于这些类型，那么采用这种语法而非通用表达式树会更有优势，因为其规则更易于可视化，也更便于进行静态分析。

尽管具有这些优点，在工程实践中应用图变换仍面临困难[2, 27]。文中引用的综述讨论了通过图重写进行计算的各种策略。一种直接的方法是无序图重写，即以任意顺序应用重写规则，也可以附加约束；然而，许多应用需要能够以系统化、特定领域的方式执行一系列原子重写，例如遍历一组匹配项。为此，一些较早的图示语言基于有向图 [5, 8], , 其中顶点是重写规则，边带有 $\mathbb{B}$ 值，表示当前规则匹配或不匹配时下一步该转向何处。还有一些表达能力更强的语言，为图程序提出了 BNF 文法[20, 19]。这类程序的控制流隐含在赋予 `while` 等构造的语义中，而不是显式表示（例如，

图的循环）。当前用于图重写编程的抽象方式，使工程师难以协作，因为不同软件实现之间，或重写规则所预设的本体之间存在不兼容。此外，对于重写中使用<sty le id='1'>哪个 匹配项的细粒度控制，也未受到重视。

我们将展示如何为一大类数据变换赋予对称幺半范畴的结构，并为其配备一个猜想性的迹算子，从而使得可以将带反馈环的连线图用作图形语法。在抽象地描述这类数据变换之后，我们将展示这一形式体系如何指导用户界面与软件实现，以构造由重写规则组合而成的复杂计算。

在第 2 节中，我们逐步构造一个范畴$\mathbf{DK}_t$，它由多项式单子 t 参数化，足以满足我们的重写应用需求。在第 3 节中，我们证明它是一个以$\mathbf{Cat}^\sharp$ 为富集范畴的余笛卡尔幺半范畴。我们还为满足某个判据的单子提出一个猜想性的迹算子。在第 4 节中，我们构造一种领域专用语言（DSL），其语义位于$\mathbf{DK}_t$ 中，用于编写通过重写规则操纵数据的复杂程序。最后，我们在第 5 节总结全文并展望未来工作。

符号表示

我们假设读者熟悉范畴、函子和富化范畴论。关于 **Poly** 及其多种幺半结构（$+, \times, \otimes, \triangleleft$）的许多技术细节可参见 [25]；此处略去这些内容，转而在第 2 节为这些构造提供直观解释。相比之下，第 3 节需要读者对 **Poly** 有扎实的技术理解。我们用 $[-,-]$ 表示范畴中的内 Hom，而用 $(-,-)$ 表示具有（余）积的范畴中的（余）配对态射。我们用 $\sum$ 表示集合的不交并。我们通常直接用对象名称表示该对象上的恒等态射，例如用 A 表示 id_A。

2 背景

本节将逐步改进用于建模我们在重写中所需类型的动力系统的框架，并提出一系列范畴：**Meal**、**Meal**/∼、**DynMeal**，以及最后的 $\mathbf{DK}_t$。每种形式体系都应允许我们将大型系统视为较小系统的复合；在任意粒度层次上，我们都可以把局部系统看作封闭的盒子，可以进入和离开，并且会随着我们与之交互而演化。

2.1 Mealy 机

我们首先考虑米利机范畴 **Meal**：

$$\mathrm{Ob}(\mathbf{Meal}) := \mathrm{Ob}(\mathbf{Set})$$
$$\mathrm{Hom}_{\mathbf{Meal}}(A,B) := \left\{(S : \mathbf{Set}, s_0 : S, \phi\colon A \times S \to S \times B)\right\}$$

一个映射可以称为动态函数。它包含一个状态集合 S 和一个特定状态 s_0。此外，给定任意状态 $s : S$，它会提供一个函数 $A \to B$，并且对于任意输入 $a : A$，还会提供一个更新后的状态；这些都编码在 $\phi(-, s)$: $A \to S \times B$ 中。因此，我们可以将 **Meal** 中的态射视为函数 $A \to B$，它每次接收到输入时都会更新。图 1a-b 展示了一个玩具模型中的态射，该模型描绘国际象棋棋子躲避陷阱。

我们声称 **Meal** 是一个对称幺半范畴，其幺半单位为 $\varnothing$，对象 $A \otimes B$ 由集合的不交并给出。对于态射的张量积和复合，我们定义[1]

$$(S, s_0, \phi) \otimes (T, t_0, \psi) := (S \times T, (s_0, t_0), \phi \times \psi)$$
$$(S, s_0, \phi) \fatsemi (T, t_0, \psi) := (S \times T, (s_0, t_0), (\phi \times T) \fatsemi (S \times \psi))$$

由于 **Meal** 是一个对 称幺半范畴，因此有一种现成的图形语言可用于可视化
串行与并行 C 其态射的复合。该语言的图标是导线，每条导线都标注有

[1]我们在记号“$\phi \times \psi$”中省略了对称性 $B \times S \cong S \times B$。

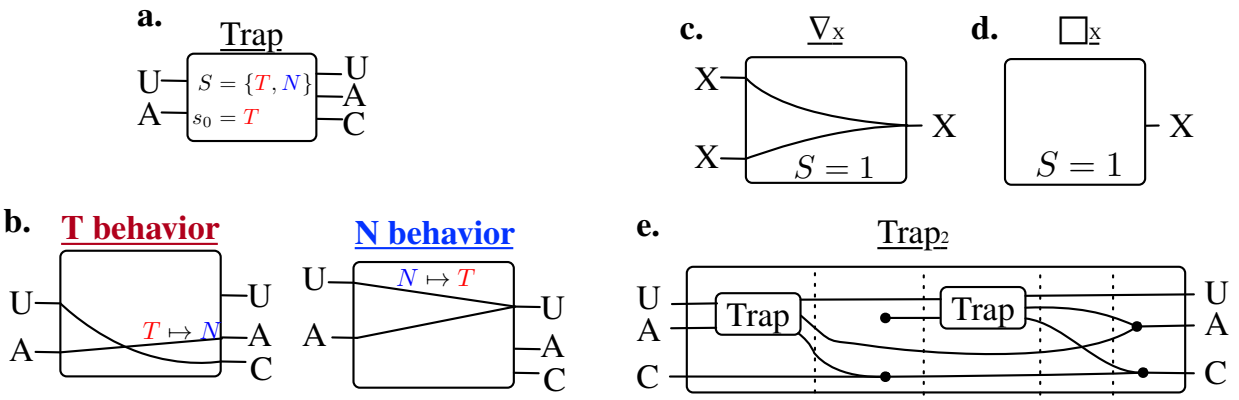


图 1：假设我们玩一种类似国际象棋的游戏，其中某些棋子可能处于不同状态：未察觉、警觉和被捕获。a. 态射 $\mathrm{Trap} : \mathrm{Hom}_{\mathbf{Meal}}(\{U, A\}, \{U, A, C\})$ 表示这样一个系统：对手可能设下陷阱来捕获棋子。这是一个动力系统，接收一个棋子（未察觉或警觉），并输出一个未察觉、警觉或被捕获的棋子。Trap 有两种可能的状态：设有陷阱（T）或未设陷阱（N），系统从状态 T 开始。b. Trap 需要一个函数 $S \times \{U, A\} \to S \times \{U, A, C\}$，通过展示两种可能状态下的动力学来将其可视化。在状态 T 中，未察觉的棋子会落入陷阱并被捕获，而警觉的棋子仍保持警觉（且状态变为 N）。在状态 N 中，所有输入都会以未察觉状态离开，但一个 U 输入会触发状态变为 T。c. Meal 中一族形如 $\mathrm{Hom}_{\mathbf{Meal}}(X + X, X)$ 的态射，其中 X 可以是任意集合。我们将其可视化为一个带有多条输入线的点。d. Meal 中一族形如 $\mathrm{Hom}_{\mathbf{Meal}}(\varnothing, X)$ 的态射，其中 X 可以是任意集合。我们将其可视化为一个带有一条输出线的点。e. 一个复合动力系统 Trap_2，描绘棋子依次通过两个 Trap 系统：从第一个子系统离开的警觉棋子会跳过第二个系统。该图正式表示五个态射的顺序复合，虚线竖线将它们区分开来：$\mathrm{Trap} \fatsemi (U + \square_A + A + \nabla_C) \fatsemi (\mathrm{Trap} + A + C) \fatsemi (U + A + \sigma_{C,A} + C) \fatsemi (U + \nabla_A + \nabla C)$。复合系统的状态空间为 $\{T, N\} \times \{T, N\}$，初始状态为 (T, T)。

一个集合、带有输入端口和输出端口的方框（也都标注有一个集合），以及编织；这些内容将在 [21, §3.1、3.5] 中详细介绍。简而言之，数据沿着作为对象的导线从左向右流动。垂直维度表示对象和态射的张量积。我们将 $\mathrm{Hom}_{\mathbf{Meal}}(A, B)$ 中的一个元素表示为一个带有 $|A|$ 个输入端口和 $|B|$ 个输出端口的方框。图 1e 给出了一个非平凡态射复合的图示示例。

我们最终的目标是构造一个带余积的带迹幺半范畴；然而，Meal 中的余积通常并不存在。对于态射 $(S, s_0, \phi) : \mathrm{Hom}_{\mathbf{Meal}}(A, Z)$ 和 $(S', s'_0, \psi) : \mathrm{Hom}_{\mathbf{Meal}}(B, Z)$，自然的配对方式是将两个方框并排放置并合并其输出，即 $(S, s_0, \phi) \otimes (S, s_0, \phi) \fatsemi \nabla_Z$，其中 $\nabla_Z \colon Z + Z \to Z$ 是余对角态射。这种做法在某种意义上具有正确的行为，具体将在下一节说明。然而，所要求的余积等式（例如 $\iota_1 \fatsemi (\phi, \psi) = \phi$）通常并不成立，因为状态空间并不相同，即 $S \times S' \cong^? S$。

2.2 按行为等价关系取商

现在，我们将行为本身作为正式的研究对象，而不再仅仅将其视为由米利机诱导而来。尽管输入为 A、输出为 B 的米利机的行为集合，可以简洁地刻画为 $X \mapsto B^A X^A$ 上终余代数的底层集合——我们最终将其记作 $\mathfrak{c}\langle [Ay, By] \rangle(1)$——但从多项式函子的视角看待行为，能让我们更自然地推广结果；事实上，上述记号正是由此而来：$[Ay, By]$ 是 **Poly** 中的内 Hom，而 $\mathfrak{c}\langle - \rangle$ 是 **Poly** 上的余自由余幺半群构造。

多项式函子在形式上是可表示函子 **Set** $\to$ **Set** 的和，也就是说，形如 $y^A := \mathrm{Hom}_{\mathbf{Set}}(A, -) : \mathbf{Set} \to \mathbf{Set}$ 的函子，将任意集合 X 映射为 X^A。这个和由集合 I 索引，因此多项式可记作 $p := \sum_{i:I} y^{A_i}$。注意，I 与 $p(1)$ 典范同构。$p(1)$ 的元素称为位置，而对于每个位置 $i : p(1)$，都有一个方向集合 A_i。我们将 A_i 记作 $p[i]$，因此

我们可以写作 $p \cong \sum_{i:\ p\ (1)} y^{p[i]}$。多项式函子可以视为系统接口，下面展示了几种表示方式。

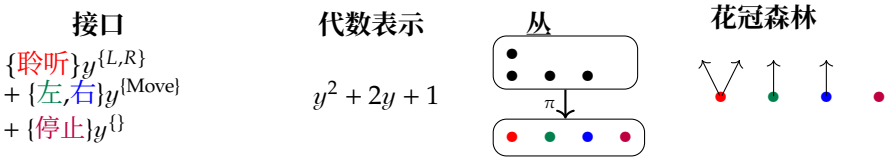


图2：刻画一种接口的多项式函子的各种表示。该接口可以执行三种操作：监听（并接收一个布尔值）、移动（向左或向右），或者停止。

多项式余单子可以与范畴 [1] 以及余函子相对应，我们将这个范畴记作 **Cat**$^\sharp$。余自由余单子函子 $\mathfrak{c}\langle -\rangle$: **Poly** $\to$ **Cat**$^\sharp$ 可以理解为将一个多项式 p 映射到一个范畴；该范畴的对象是具有接口 p的系统可能产生的行为树。这些树可能是无限的，其构造方式是从一个根节点开始，逐层叠加单步行为，如多项式的花冠森林表示所示。为了帮助理解这些行为树的形态，图3展示了与图1a-b中的米利机对应的一棵树，以及与上述示例接口对应的一棵树。由此，我们可以定义范畴 **Meal**/$\sim$:

$$\begin{aligned} \mathrm{Ob}(\mathbf{Meal}/\sim) &:= \mathrm{Ob}(\mathbf{Set}) \\ \mathrm{Hom}_{\mathbf{Meal}/\sim}(A, B) &:= \mathrm{Ob}\ \mathfrak{c}\langle [Ay, By] \rangle \end{aligned} \tag{1}$$

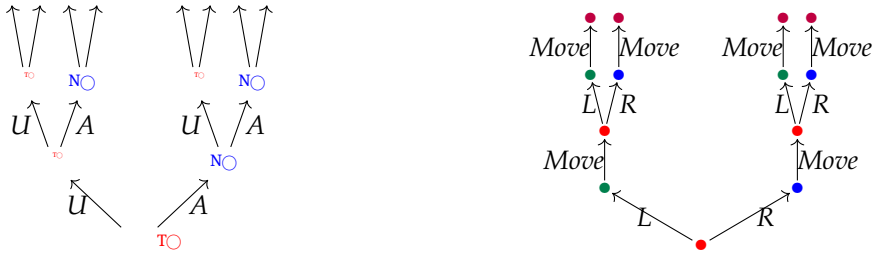


图3：**左**。一个 $\mathfrak{c}\langle [\{U,\ A\}y,\ \{U,\ A,\ C\}y] \rangle$，它是 陷阱 在图1b中的行为。注意 $[2y, 3y] \cong 3^2y^2$，也就是说，它是一个接口，接收三种可能输入之一，并且可以配置为任意可能的函数 $2 \to 3$。我们将这八种可能的函数中的两种标记为 T 和 N。**右**。图2接口的一种可能行为，即范畴 $\mathfrak{c}\langle y^2 + 2y + 1 \rangle$的一个对象。机器从监听状态开始，收到输入 L (分别是 R)后向左（分别向右）移动。机器再重复一次，然后停止。

有一个重要特征尚未得到显式建模。Hom(A, B) 只是一个由行为构成的集合，但在系统随时间演化时，我们需要利用这个行为集合所具有的丰富结构：系统随着接收到新的输入而发生变化的方式。对我们的范畴进行富化，即将每个态射集合替换为一个范畴，就能让我们刻画态射如何随着输入的接收而变化。

2.3 对系统随时间演化的建模

回顾一下，$\mathfrak{c}\langle p \rangle$ 将多项式 p 映射到一个范畴。到目前为止，我们只考虑了该范畴的对象。态射树$_1 \to$ 树$_2$ 由树$_1$, 中的一条路径给出，该路径从根节点出发，终止于树$_2$ 的一个副本处（见图 4）。这对于显式建模输入序列如何导向一个新的动力系统至关重要。我们定义**DynMeal**，使其具有与**Meal** 相同的对象，但富化于**Cat**$^\sharp$ 而非**Set**，使得同态对象 $\mathrm{Hom}^{\mathbf{DynMeal}}(A, B) := \mathfrak{c}\langle [Ay, By] \rangle$。

范畴 **Cat**$^\sharp$ 是 **Poly** 中的 $\multimap$余幺半群范畴；正如 [1, 14] 中所证明的，它等价于范畴与余函子构成的范畴。余函子在对象上有一个正向映射，但在态射上有一个反向映射。这意味着，对于复合系统 $\phi \cdot \psi$，我们可以构造一个复合

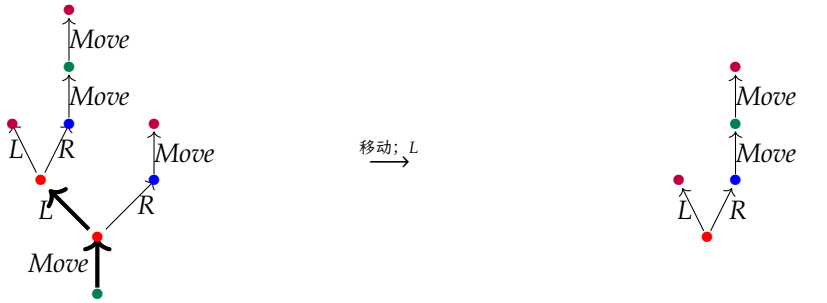
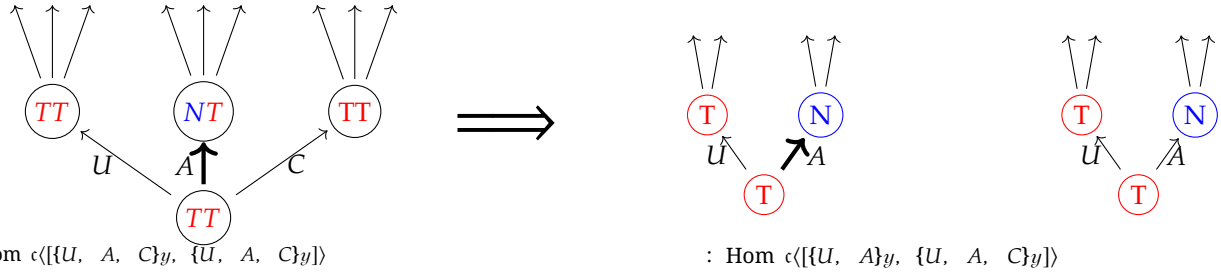


图 4： $\mathfrak{c}\langle y^2+2y+1\rangle$中两个对象之间的一个态射，态射的数据用粗箭头表示，即移动，然后 L。

行为树：给定 ϕ 和 ψ的行为树，但 ϕ 和 ψ 随时间的演化方式由$\phi\cdot\psi$ 随时间的演化方式决定，例如参见图 5。**Poly** 中的每个余幺半群都有一个底层多项式，且与之相关联的函子 U 有一个右伴随 $\mathfrak{c}\langle-\rangle$：

$$\mathbf{Poly} \underset{U}{\overset{\mathfrak{c}\langle-\rangle}{\underset{\longleftarrow}{\overset{\longrightarrow}{\top}}}} \mathbf{Cat}^\sharp \tag{2}$$



： Hom $\mathfrak{c}\langle[\{U,\ A,\ C\}y,\ \{U,\ A,\ C\}y]\rangle$

： Hom $\mathfrak{c}\langle[\{U,\ A\}y,\ \{U,\ A,\ C\}y]\rangle$

图5：复合系统 Trap$_2$（图1e）响应输入A时行为变化的示例。与图3中一样，T 和 N 表示 3^2, 的元素，标签 TT 和 NT 指的是 3^3, 中由 Trap$_2$ 发送其输入的位置所诱导的元素；布线图的数据 Trap$_2$ 说明如何将这种 A-交互转换为其 Trap 子组件的交互。请注意，对于 A 输入，第二个 Trap 子组件完全没有更新，即其更新是 $\mathfrak{c}\langle[\{U,A\}y,\ \{U,A,C\}y]\rangle$中的恒等态射。

这个故事与余代数的故事有着密切联系。对于任意多项式 p，$\mathfrak{c}\langle p\rangle$对象的集合与 $X\mapsto B^AX^A$ 上终余代数的底层集合同构。函子$\mathfrak{c}\langle p\rangle\to\mathbf{Set}$ 构成的范畴等价于 p-余代数范畴。每棵行为树在 $\mathfrak{c}\langle p\rangle$ 中都映射到具有该行为的状态集合。

DynMeal 的富化结构现在明确展现了我们的行为树如何随输入而变化。最后一项改进是提升表达能力，例如增加进入某个环境后永远无法退出的可能性，考虑可能结果的列表，以及考虑可能结果的概率分布。

2.4 添加单子效应

通过将 Set 上的多项式单子（t，η，μ）纳入我们的态射，可以大幅拓展迄今所具备的表达能力。值得关注的单子有

$$\text{Maybe}=y+1,\qquad \text{Writer}_M=My,\qquad \text{List}=\sum_{N:\mathbb{N}}y^N,\qquad \text{and Dist}=\sum_{N:\mathbb{N}}\Delta_N y^N \tag{3}$$

其中 M 是一个幺半群，且 $\Delta_N=\{P\colon N\to[0,1]\mid 1=\sum P(i)\}$。迄今为止，我们的工作针对的是一般的多项式，而不仅仅是形如 $Ay\to By$的接口，因此可以将接口 $A\to B$ 的 Hom 对象视为 $\mathfrak{c}\langle[Ay,t\triangleleft By]\rangle$，从而将单子效应纳入我们的范畴。

定义 1（动态 Kleisli 范畴）。给定一个多项式单子 t，我们定义一个以$\mathbf{Cat}^\sharp$为富化的范畴，记作$\mathbf{DK}_t$，如下：

$$\mathrm{Ob}(\mathbf{DK}_t) := \mathrm{Ob}(\mathbf{Set})$$
$$\mathrm{Hom}_{\mathbf{DK}_t}(A, B) := \mathfrak{c}\langle [Ay, t \triangleleft By] \rangle$$

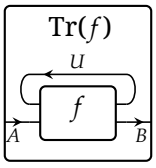
3 $\mathbf{DK}_t$ 作为带迹幺半范畴

在本节中，我们始终假设 $(t,\ \eta,\ \mu)$ 是一个笛卡尔多项式单子。这足以证明 $\mathbf{DK}_t$ 是一个余笛卡尔幺半 $\mathbf{Cat}^\sharp$-富化范畴。在第 3.1 节中，我们将证明它是一个 $\mathbf{Cat}^\sharp$-富化范畴；在第 3.2 节中，我们将证明它具有余积。

事实上，我们希望 $\mathbf{DK}_t$ 是带迹幺半的，这意味着存在一些态射

$$\mathrm{Tr}^U_{A,B} \colon \mathrm{Hom}_{\mathbf{DK}_t}(A \otimes U, B \otimes U) \to \mathrm{Hom}_{\mathbf{DK}_t}(A, B) \tag{4}$$

它们满足各种相容性条件。带迹幺半范畴具有一种图形语法，其中包括环



（另见 [21]）。这一概念最初是在 [11], 中针对 **Set**-富化范畴定义的；将其推广到 V-富化情形时，可以要求迹映射 (4) 是 V 中的一个映射。在第 3.3 节中，只要 t 按定义 2 的意义是例外的，我们就会为$\mathbf{DK}_t$提出一个迹映射。

3.1 $\mathbf{DK}_t$ 上的 $\mathbf{Cat}^\sharp$-富化范畴结构

本节假定读者对 **Poly** 相当熟悉。定义 1 给出了 $\mathbf{DK}_t$ 上拟议的 $\mathbf{Cat}^\sharp$-富化范畴结构；本节的首要目标是证明它满足恰当的性质。在第 3.2 节中，我们将证明它是幺半的。

对于任意松幺半函子 $\mathcal{V} \to \mathcal{W}$，都存在一个诱导函子 $\mathcal{V}$-**Cat** $\to \mathcal{W}$-**Cat**。对于所有幺半范畴 $(\mathcal{V}, I, \otimes)$，函子 $\mathcal{V}(I, -)$: $\mathcal{V} \to$ **Set** 是松幺半的；我们称诱导函子 V-**Cat** $\to$**Cat** 为底层范畴函子。当存在一个 $\mathcal{V}$- 范畴以 $\mathcal{C}$ 为底层范畴时，我们称范畴 $\mathcal{C}$ 富化于 $\mathcal{V}$。本节的第二个目标是证明通常的 Kleisli 范畴 $\mathbf{Set}_t$ 富化于 $(\mathbf{Cat}^\sharp, y, \otimes)$。

我们的首要策略是证明 $\mathbf{Set}_t$ 在（**Poly**，y，$\otimes$）中带有富化结构。

定理 1. 设 (t, η, μ) 是一个多项式单子。*Kleisli* 范畴 $\mathbf{Set}_t$ 在 $(\mathbf{Poly}, y, \otimes)$中带有富化结构。

证明。对于集合 A, B : **Set**，定义多项式

$$h^t_{A,B} := [Ay, t \triangleleft By]. \tag{5}$$

形如 $y \to h^t_{A,B}$ 的映射与多项式映射 $Ay \to t \triangleleft By$ 一一对应。注意，这些映射又与函数 $A \to t \triangleleft B$ 一一对应，而它们恰好就是 Kleisli 态射，即我们有

$$\mathbf{Poly}(y, h^t A, B) \cong \mathbf{Set}_t(A, B). \tag{6}$$

因此，只需为同态对象 $h^t_{A,B}$: **Poly**定义恒等态射，以及含单位元的结合复合法则。

A 上的恒等态射，我们记作

$$Ay \cong y \triangleleft Ay \xrightarrow{\eta \triangleleft Ay} t \triangleleft Ay.$$

形如$\overset{t}{h}_{A,\ B} \otimes \overset{t}{h}_{B,\ C} \to \overset{t}{h}_{A,\ C}$的映射与映射 $Ay \otimes [Ay, t \triangleleft By] \otimes [By, t \triangleleft Cy] \to Cy$ 一一对应，因此从求值映射 $Ay \triangleleft [Ay, t \triangleleft By] \to t \triangleleft By$ 开始，只需找到一个映射 $(t \triangleleft By) \otimes [By, t \triangleleft Cy] \to Cy$。由于 **Poly** 是双幺半范畴，我们得到所需的映射：

$$(t \triangleleft By) \otimes (y \triangleleft [By, t \triangleleft Cy]) \xrightarrow{\text{duoid}} (t \otimes y) \triangleleft (By \otimes [By, t \triangleleft Cy]) \xrightarrow{\text{eval}} t \triangleleft t \triangleleft Cy \xrightarrow{\mu} t \triangleleft C.$$

容易验证，这些定义是结合的且含单位元。 □

推论 1. 设 (t, η, μ) 是一个多项式单子。*Kleisli* 范畴 $\mathbf{Set}_t$ 在 $(\mathbf{Cat}^\sharp, y, \otimes)$中富集。

证明。 对于集合 A, B，令 $\overset{t}{h}_{A,B} : \mathbf{Poly}$ 如式 (5) 所示。由于函子 $\mathfrak{c}\langle - \rangle : \mathbf{Poly} \to \mathbf{Cat}^\sharp$ 是松幺半群的，我们可以定义同态对象

$$\mathbf{DK}_t(A, B) \coloneqq \mathfrak{c}\langle h^t_{A,B} \rangle = \mathfrak{c}\langle [Ay, t \triangleleft By] \rangle$$

从而得到一个 $\mathbf{Cat}^\sharp$-范畴。为证明其底层范畴是 $\mathbf{Set}_t$，我们使用等式 (2) 和 (6)

$$\mathbf{Cat}^\sharp(y, \mathfrak{c}\langle h^t_{A,B} \rangle) \cong \mathbf{Poly}(y, h^t_{A,B}) \cong \mathbf{Set}_t(A, B),$$

至此证明完毕。 □

作为一个$\mathbf{Cat}^\sharp$-范畴，$\mathbf{DK}_t$ 不仅具有由松幺半函子$\mathbf{Cat}^\sharp(y, -) : \mathbf{Cat}^\sharp \to \mathbf{Set}$诱导的底层普通范畴，也具有由松幺半函子 Ob: $\mathbf{Cat}^\sharp \to \mathbf{Set}$诱导的另一个相应普通范畴。这两个范畴之间存在一个自然变换：

$$(\mathbf{Cat}^\sharp, y, \otimes) \underset{\text{Ob}}{\overset{\mathbf{Cat}^\sharp(y,-)}{\Downarrow}} (\mathbf{Set}, 1, \times)$$

根据定义（1），我们通过在局部应用 Ob，从$\mathbf{DK}_t$ 得到$\mathbf{Meal}/\sim$，如下所示。**命题**

1. 由 $\mathrm{Ob} : \mathbf{Cat}^\sharp \to \mathbf{Set}$ 诱导的函子$\mathbf{Cat}^\sharp\text{-}\mathbf{Cat} \to \mathbf{Set}\text{-}\mathbf{Cat}$ 将$\mathbf{DK}_t \mapsto \mathbf{Meal}/\sim$.

3.2 $\mathbf{DK}_t$上的余积幺半结构

接下来我们证明$\mathbf{Cat}^\sharp$-范畴$\mathbf{DK}_t$ 具有余积，其定义见 [12, §3.8] ，详述见 [23]。我们需要证明，对于任意两个集合 $A, B : \mathbf{Set}$，存在态射

$$i \colon y \to \mathfrak{c}\langle h^t_{A,A+B} \rangle \qquad \text{and} \qquad j \colon y \to \mathfrak{c}\langle h^t_{B,A+B} \rangle$$

使得对于每个 $C : \mathbf{Set}$、每个 $v : \mathbf{Cat}^\sharp$ 以及每对映射 $f \colon v \to \mathfrak{c}\langle \overset{t}{h}_{A,C} \rangle$ 和 $g \colon v \to \mathfrak{c}\langle \overset{t}{h}_{B,C} \rangle$，都存在唯一的态射 $(f, g) \colon v \to \mathfrak{c}\langle \overset{t}{h}_{A+B,C} \rangle$，使得下列图交换

$$\begin{array}{ccc} y \otimes v & \xrightarrow{i \otimes (f,g)} & \mathfrak{c}\langle h^t_{A,A+B} \rangle \otimes \mathfrak{c}\langle h^t_{A+B,C} \rangle \\ \cong \downarrow & & \downarrow \wr \\ v & \xrightarrow[f]{} & \mathfrak{c}\langle h^t_{A,C} \rangle \end{array} \qquad \begin{array}{ccc} y \otimes v & \xrightarrow{j \otimes (f,g)} & \mathfrak{c}\langle h^t_{B,A+B} \rangle \otimes \mathfrak{c}\langle h^t_{A+B,C} \rangle \\ \cong \downarrow & & \downarrow \wr \\ v & \xrightarrow[g]{} & \mathfrak{c}\langle h^t_{B,C} \rangle \end{array} \tag{7}$$

定理 2。 对于任意多项式单子$(t、\ \eta、\ \mu)$，$\mathbf{Cat}^\sharp$-范畴$\mathbf{DK}_t$ 具有余积。

证明。 对于任意 $p、\ q、\ x$： **Poly**，这些包含态射 $p \to p + q \leftarrow q$ 诱导出一个同构

$$[p + q, x] \xrightarrow{\cong} [p, x] \times [q, x]$$

在**Poly**中，因此特别地有

$$h^t_{(A+B),C} \cong h^t_{A,C} \times h^t_{B,C}.$$

根据 (2)， $v \to \mathfrak{c}\langle p \rangle$ $\mathbf{Cat}^\sharp$ 中的映射与多项式的映射 $v \to p$ 一一对应。因此，我们可以将一对 $\mathbf{Cat}^\sharp$-映射 $f \colon v \to \mathfrak{c}\langle \overset{t}{h}_{A,C} \rangle$ 和 $g \colon v \to \mathfrak{c}\langle \overset{t}{h}_{B,C} \rangle$ 视为一个 **Poly**-映射 $(f, g) \colon v \to \overset{t}{h}_{A+B,C}$。 (7) 中图的交换性很容易得出。□

3.3 $\mathbf{DK}_t$上的拟迹结构

本节为某些多项式单子 t 提出 $\mathbf{DK}_t$ 上的一种迹结构。我们称它们为例外的，因为它们带有一个元素 $\xi\colon 1 \to t$，其作用类似于异常：如果语法树的任何分支抛出异常，那么整棵树也会如此。

定义迹映射需要比前几节更多地了解**Poly** 。第 3.1 节和第 3.2 节中定义的$\mathbf{Cat}^\sharp$-范畴结构在$\mathbf{DK}_t$上是由更简单的**Poly**-范畴结构在$\mathbf{DK}_t$上诱导而来的；而应用于重写以及迹结构时，都需要用到完整的$\mathbf{Cat}^\sharp$-富化结构。

我们的首要目标是定义异常单子。

引理 1. 对于任意笛卡尔多项式单子 (t, η, μ)，多项式 $t+1$ 也具有多项式单子的结构，并且余积包含 $t \to t+1$ 是一个单子态射。

证明。 只需证明，对于任意 t，存在一个分配律 $t \triangleleft (y+1) \to (y+1) \triangleleft t$，它与两侧的包含映射 $y \to y+1$ 相交换。

在任何具有终对象的范畴中，余积包含 $A \rightarrowtail B$，即对于某个 A'，有同构 $A + A' \cong B$，诱导出一个映射

$$B \xrightarrow{\cong} A + A' \xrightarrow{A+!} A + 1$$

使得 $A \to B \to A+1$ 是余积包含。

容易证明，**Poly** 中的每个笛卡尔单态射都是余积包含。此外，对于任意多项式 p，映射 $p \cong p \triangleleft y \to p \triangleleft (y+1)$ 是笛卡尔单态射，因为 $y \to y+1$ 是，并且 $\triangleleft$在两个变量中都保持单态射和笛卡尔映射。因此，我们得到一个映射 $p \triangleleft (y+1) \to p+1 \cong (y+1) \triangleleft p$，它关于笛卡尔映射 $p \to p'$ 是自然的。可以验证，当 t 是一个笛卡尔单子时，该映射始终是一个分配律，从而完成证明。□

定义 2（异常单子）。设 (t, η, μ) 是一个笛卡尔多项式单子。t 上的异常 结构是一个截面 $\xi\colon t+1 \to t$，对应于引理 1 中的单子包含映射 $t \to t+1$。

根据引理 1，对于任意笛卡尔多项式单子 t，$t+1$ 都是一个异常单子。

注 1。在异常单子的乘法 $\mu\colon t \triangleleft t \to t$ 中，左侧的一个位置由一个位置 $I\colon t(1)$ 以及对每个方向 $i\colon t[I]$，一个位置 $J_i\colon t(1)$ 组成。如果 I 或任何一个 J_i 是异常元素 ξ，那么它在 μ 下的像必定是异常元素：任何地方出现异常，都会导致整个计算出现异常。

异常单子的概念不同于带零元的单子 [28]，尽管在这两种情况下，单子 t 都配备了一个常量 $1 \to t$。例如，列表单子有一个零元，即空列表，但它并不是异常单子，因为列表的列表可以包含空列表，而其串接结果却不为空。

接下来，我们提出迹映射 $\mathrm{Tr}^U_{A,B}\colon \mathfrak{c}\langle[(A+U)y,\ t \triangleleft (B+U)y]\rangle \to \mathfrak{c}\langle[Ay, t \triangleleft By]\rangle$。事实上，按照 [21]，定义一个形式如下的迭代映射会更直接：

$$iter^A_B\colon \mathfrak{c}\langle[Ay, t \triangleleft (A+B)y]\rangle \to \mathfrak{c}\langle[Ay, t \triangleleft By]\rangle \tag{8}$$

此时，我们可以将 $\mathrm{Tr}^U_{A,B}$ 定义为以下复合：

$$\begin{aligned}\mathfrak{c}\langle[(A+U)y, t \triangleleft (B+U)y]\rangle &\to \mathfrak{c}\langle[(A+U)y, t \triangleleft (A+U+B)y]\rangle \\ &\xrightarrow{iter^A_B} \mathfrak{c}\langle[(A+U)y, t \triangleleft By]\rangle \\ &\to \mathfrak{c}\langle[Ay, t \triangleleft By]\rangle.\end{aligned}$$

为此，我们需要更明确地说明多项式 p 上的余自由余单子构造和自由单子构造。

带基点多项式 $y \to p$ 上的自由单子 $\mathfrak{m}\langle p\rangle$ 分两步构造。对于任意有限带基点多项式 q——即每个 $q[J]$ 都是有限集—— q 上的自由单子可以一步构造，即取其余极限：

$$\mathfrak{m}\langle q\rangle := \mathrm{colim}(\cdots \leftarrow q^{\triangleleft n+1} \xleftarrow{g_n} q^{\triangleleft n} \leftarrow \cdots \leftarrow q)$$

其中映射 $g_n\colon q^{\triangleleft n} \to q^{\triangleleft n+1}$ 由 $p^{\triangleleft n} \cong y \triangleleft p^{\triangleleft n} \to p \triangleleft p^{\triangleleft n} = p^{\triangleleft n+1}$归纳定义。任意多项式 p 都可以表示为其垂直投影 $p \to p^j$ 的滤过极限，这些投影指向有限多项式：也就是说，对于每个和分量 $I\colon p(1)$，只需将其中的 $p[I]$ 替换为任意有限子集，并对所有此类对象在逐分量投影下取极限。该极限与 p同构，我们记作$p \cong \lim_{j:J_p} p^{(j)}$。根据构造，这些 $p^{(j)}$ 都是有限的，因此令 $\mathfrak{m}\langle p^{(j)}\rangle \coloneqq \mathfrak{m}\langle p^{(j)}\rangle$ 表示其上的自由单子，按上述方式构造。最后，我们将 p 上的自由单子 $\mathfrak{m}\langle p\rangle$ 构造为它们的滤过极限：

$$\mathfrak{m}\langle p\rangle \coloneqq \lim_{j:J_p} \mathfrak{m}\langle p^{(j)}\rangle.$$

余幺半群 $\mathfrak{c}\langle p\rangle$ 在 p 上的构造：**Poly** 只需一步即可完成。它由极限承载

$$\mathfrak{c}\langle p\rangle \coloneqq \lim(\cdots \to p^{(n+1)} \xrightarrow{f^{(n)}} p^{(n)} \to \cdots \to p^{(1)} \xrightarrow{f^{(0)}} p^{(0)})$$

其中， $p^{(k)}$ 按如下方式归纳定义：

$$p^{(0)} \coloneqq y \qquad\qquad p^{(k+1)} \coloneqq (p \triangleleft p^{(k)}) \times y$$

映射 $f_{(k)}\colon p^{(k+1)} \to p^{(k)}$ 按如下方式归纳定义：

$$p^{(0)} = p \times y \xrightarrow{f^{(0)}:=\mathrm{proj}} y = p^{(0)} \qquad p^{(k+1)} = (p \triangleleft p^{(k+1)}) \times y \xrightarrow{f^{(k+1)}:=(p\triangleleft f^{(k)})\times y} (p \triangleleft p^{(k)}) \times y = p^{(k+1)}$$

命题 2. 设 $y \to p$ 是多项式的一个笛卡尔映射，且 $\mathfrak{m}\langle p\rangle$ 是其上的自由单子；设 $\mathfrak{c}\langle p\rangle$是 p上的余自由余单子。存在一个自然笛卡尔单态射

$$\mathfrak{m}\langle p\rangle \to \mathfrak{c}\langle p\rangle \triangleleft (y+1).$$

证明。 对于任意 $q\colon$ **Poly**，函子 $-\triangleleft q$与极限相容。由于 $\mathfrak{c}\langle -\rangle$是右伴随，它也与极限相容。由于笛卡尔单态射的极限仍是笛卡尔单态射，我们可以假设p 是有限的；因此，只需对每个 $i\colon \mathbb{N}$构造相容的笛卡尔单态射 $\varphi_i\colon p^{\triangleleft i} \to p^{(i)} \triangleleft (y+1)$。我们取 φ_0 为笛卡尔单态射 $y \to y+1$。设给定 φ_i。要定义 $p^{\triangleleft i+1} \to (p \triangleleft p^{(i)}) \triangleleft (y+1)$，只需给出两个映射： $p^{\triangleleft i+1} \to p \triangleleft p^{(i)} \triangleleft (y+1)$和 $p^{\triangleleft i+1} \to y+1$。对于后者，使用 $p_{(i+1)} \to 1 \to y+1$。还需给出一个笛卡尔单态射 $p^{\triangleleft i+1} \to p \triangleleft p^{(i)} \triangleleft (y+1)$，我们通过归纳 $p \triangleleft p^{\triangleleft i} \xrightarrow{p\triangleleft\varphi_i} p \triangleleft p^{(i)} \to p \triangleleft p^{(i)} \triangleleft (y+1)$得到它。 □

引理 2。 在**Poly**中，任意笛卡尔单态射 $p \to q$ 都是余积包含。

证明。 设 $\varphi\colon\ p \to q$ 是一个笛卡尔单态射。由于 $p \mapsto p(1)$ 是一个右伴随， $\varphi(1)\colon\ p(1) \to q(1)$ 是一个注入；令 $J' \coloneqq q(1) - p(1)$ 为其补集。于是我们得到所需的同构

$$q \cong p + \sum_{j:J'} y^{q[j]}.$$

□

命题 3。 对于任意带基点多项式 $y \to p$，存在一个映射 $\mathfrak{c}\langle p\rangle \triangleleft (y+1) \to \mathfrak{m}\langle p\rangle + 1$。

证明。 根据命题 2，我们有一个笛卡尔单态射 $\mathfrak{m}\langle p\rangle \to \mathfrak{c}\langle p\rangle \triangleleft y + 1$，由引理 2 可知，它是一个余积包含。对于具有终对象的范畴中的任意余积 $p \to q$ 包含，我们都有一个映射 $q \cong p + p' \to p + 1$，从而完成证明。□

让我们回到最初的目标：构造一个如式（8）所示的 *iter* 映射。根据（2），只需定义一个形如 $\mathfrak{c}\langle [Ay, t \triangleleft (A+B)y]\rangle \to [Ay, t \triangleleft By]$ 的多项式映射，或者等价地，定义一个形如 $Ay \otimes \mathfrak{c}\langle [Ay, t \triangleleft (A+B)y]\rangle \to t \triangleleft By$ 的映射。给定例外结构 $\xi\colon\ t+1 \to t$ on t，以及任意单子 t 都带有代数结构$\mathfrak{m}\langle t\rangle \to t$这一事实，根据命题 3，只需找到一个映射 $Ay \otimes \mathfrak{c}\langle [Ay, t \triangleleft (A+B)y]\rangle \to \mathfrak{c}\langle t\rangle \triangleleft (y+1) \triangleleft By$。

我们再次通过归纳得到所需的映射。右侧是极限

$$\mathfrak{c}\langle t\rangle \triangleleft (y+1) \triangleleft By \cong \lim(t^{(i)} \triangleleft (y+1) \triangleleft By).$$

当 $i = 0$ 时，我们使用 $Ay \otimes \mathfrak{c}\langle[Ay, t \triangleleft (A+B)y]\rangle \to 1 \to t^{(0)} \triangleleft (y+1) \triangleleft By$。设给定一个映射 φ_i：$Ay \otimes \mathfrak{c}\langle[Ay, t \triangleleft (A+B)y]\rangle \to t^{(i)} \triangleleft (y+1) \triangleleft By$。定义

$$\begin{aligned}
Ay \otimes \mathfrak{c}\langle[Ay, t \triangleleft (A+B)y]\rangle &\to y \times (Ay \otimes ([Ay, t \triangleleft (A+B)y] \triangleleft \mathfrak{c}\langle[Ay, t \triangleleft (A+B)y]\rangle) \\
&\to y \times (t \triangleleft (A+B)y \triangleleft \mathfrak{c}\langle[Ay, t \triangleleft (A+B)y]\rangle) \\
&\to y \times (t \triangleleft ((Ay \triangleleft \mathfrak{c}\langle[Ay, t \triangleleft (A+B)y]\rangle) + By)) \\
&\xrightarrow{\cong} y \times (t \triangleleft ((Ay \otimes \mathfrak{c}\langle[Ay, t \triangleleft (A+B)y]\rangle) + By)) \\
&\to y \times (t \triangleleft ((t^{(i)} \triangleleft (y+1) \triangleleft By) + By)) \\
&\to y \times (t \triangleleft t^{(i)} \triangleleft (y+1) \triangleleft By) \\
&\to (y \times (t \triangleleft t^{(i)})) \triangleleft (y+1) \triangleleft By) \\
&\xrightarrow{\cong} t^{(i+1)} \triangleleft (y+1) \triangleleft By
\end{aligned}$$

如上所述，我们现在已经定义了所谓的 *iter*映射，也就定义了迹映射 .

猜想 1. 上文定义的 $\mathbf{DK}_t$ 所谓的迹满足带迹幺半范畴的公理。

尽管这一猜想尚未得到证明，但它影响了开源软件 AlgebraicRewriting.jl 中可用实现的开发。这是第 4 节的主题。

4 重写与基于主体的建模应用

4.1 带属性的 C-集合

我们的案例研究采用 AlgebraicJulia 生态系统 [16]，因为它支持线图操作（Catlab.jl），并提供图重写库 AlgebraicRewriting.jl [4]。AlgebraicJulia 的核心数据结构是 ACSet，即针对某个有限呈示范畴 $\mathcal{C}$的带属性 $\mathcal{C}$-Set。ACSet 为数据库提供了一种范畴论模型，将 $\mathcal{C}$-Set（即余预层）扩展为可包含非组合数据 [17]。数据库模式由一个伴随函子给出，即函子 S： $|S| \to 2$，它将 $|S|$ 中的对象区分为表或属性类型。给定一个赋值 K： $S_1 \to \mathbf{Set}$，为属性提供具体的 Julia 类型后，范畴 ACSet_K^S 是双完备的，也是一个托扑斯，因此适合用于应用图重写规则。

考虑一个 ACSet X，它配备了一个指定的“焦点”，即态射 $A \to X$。我们很快会看到一些应用，在这些应用中，将 A 视为世界状态 X中特定主体的形状 是有意义的；该主体由所选态射指定。请注意，不考虑任何主体的 ACSet，等同于将主体形状设为空 ACSet 0，而它是 ACSet_K^S 的始对象。

例如，考虑这样一个建模任务：在有向图上分布狼、羊和草，其中狼和羊朝向特定方向，并拥有整数个能量单位。草生长在顶点上，并且距离长成还需整数天数。图 6 中的模式展示了对此建模的一种方式。

4.2 用于图重写程序的 ADSL

我们使用$\mathbf{DK}_t$ 理论来实现一种图重写编程语言，其中数据操作由作用于 ACSet（可能带有主体）的重写规则指定。这些程序可以由少量原始生成元组合而成。结合 Catlab 操作连线图的通用基础设施，这些原语构成了一种强大的领域专用语言，可用于基于主体的建模和通过图重写进行编程。

我们所关注的态射的定义域和值域由图 7 所示形式的图集合及其余积构成。每个这样的图——我们称之为 轨迹——都是一系列“世界状态”X_i，其中包含特选焦点 $A_i \to X_i$。如果在伪代码中用变量`t::Traj` 表示该轨迹，那么令 `last(t)` 返回 X_n，`length(t)` 返回 n，并令 `t[i]` 返回 $A_i \to X_i$。令`postcompose(t::Traj,f::Hom(X,Xi),i::Int)` 计算 `f` 与从 i 到 n 的部分映射的复合。该操作返回一个全态射或`nothing`。令 `(t::Traj)+(b::Hom(A,Xn))` 通过恒等部分映射扩展该轨迹。由于现在我们的态射的定义域和值域是无限集合，我们采用以下简写：

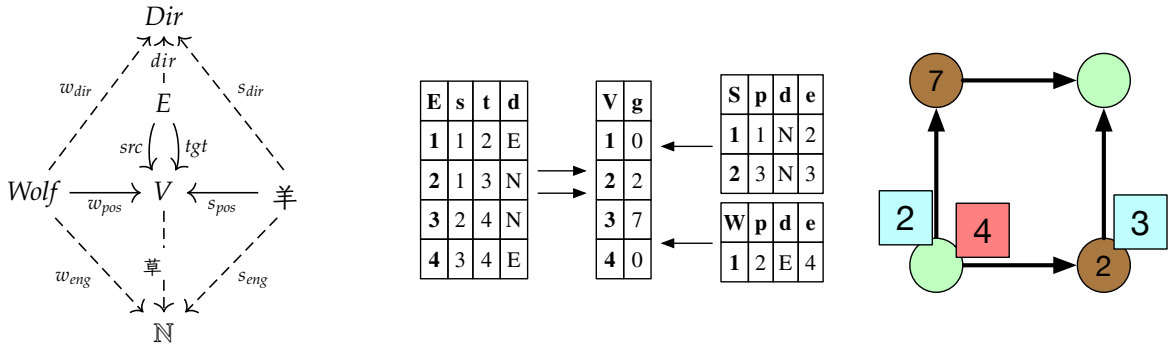


图 6：**左侧**一个 ACSet 模式，包含六个对象和态射，以及两种属性类型和六个属性（虚线边）。**中间**此模式上的一个 ACSet 示例。**右侧**该示例的非正式可视化：能量表示为整数，羊表示为蓝色方框，狼表示为红色方框。若要将狼区分为“主体”，则可使用一个 ACSet 态射，从一个包含一只狼、零条边和一个顶点的 ACSet 映射到此示例中，也就是狼的形状。

在可视化连线图时，一条标有 ACSet A_n 的连线表示当前主体为 A_n 的所有轨迹组成的集合。声明一个态射的类型为 $A + B \to C$（对于 ACSet A, B, C）表示：其定义域是当前主体为 A 的轨迹集合与当前主体为 B 的轨迹集合的余积，而值域是当前主体为 C 的轨迹集合。

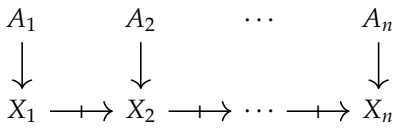
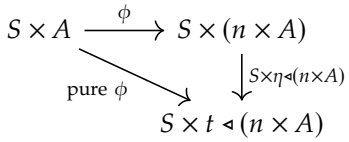


图 7：ACSet 空间中的一条轨迹。 X_1 和 X_n 分别表示模拟过程中世界的初始状态（即当前状态），每个后续世界状态都通过一个部分映射与前一状态关联（以带刻度的箭头表示）。每个世界状态 X_i 还具有一个特别指定的焦点 $A_i \to X_i$。

图 8 和图 9 展示了一些用于重写程序的生成态射。无状态生成器的语义在图 8 中以可视化方式表示，并在此说明：重写通过应用重写规则所诱导的部分映射来扩展轨迹（支持 DPO、SPO、SqPO 和 PBPO+ [15]语义）。重写规则的模式 L 和替换 R 还必须分别与特定的输入智能体形状 A 和输出智能体形状 B关联。输入智能体形状通过一个必须满足交换律的三角图，对有效匹配施加严格约束。即便如此，如果存在多个有效匹配，也会任意选择其中一个。如果重写成功，则退出 B 输出端口；否则退出 A 输出端口。`Weaken` 通过预合成主体态射来扩展轨迹，而不改变世界状态 X_n 。`Strengthen`通过推出扩展轨迹，同时改变智能体形状和世界状态。`Init` 将轨迹切换到特定的世界状态和主体，与前一个世界状态没有关联。失败可以赋予引发异常的语义。另外，如果上下文是 List + 1单子，它也可以不作提示地生成一个空列表。

`ControlFlow` 框的语义是将输入重定向到它的某个输出端口；这种重定向可能是不确定的，也可能取决于其轨迹数据。这些态射具有 $\mathrm{Hom}_{\mathbf{DK}_t}(A, n \times A)$ 的形式，其中 n 是某个集合，并且具有一个纯的米利转移函数。所谓“纯”，是指该函数可以分解为映射 $\phi : S \times A \to S \times (n \times A)$（满足 $\phi \fatsemi \pi_3 = \pi_2$），再接上单子单位 η。



Query 框的语义是一个状态空间为 List($\mathrm{Hom}_{\mathrm{ACSet}}$) $\times\mathbb{N}$ 的米利机。第一个元素是待处理的已排队“代理”列表，第二个元素则记录最初进入Query 框时（沿轨迹）的时间步。其动力学由两个函数给出：更新函数 $S \times A + C \to S$ 和读出函数 $S \times A + C \to A + B + 0$。在步骤 n 从 A 端口进入，表示重新启动Query 流程；内部状态会被覆盖，以存储n 以及一个态射列表 $B \to X_n$。从 C 端口进入表示已完成某个子代理的子程序——该主体会从框的状态中移除，所有其他主体则被推进到当前时间步。退出Query 框的方式首先取决于是否还有待处理的主体；如果有，则带着一个 B 主体从 B 端口退出。如果没有，我们会尝试带着原始的 A 主体退出。如果该主体在推进到当前状态后是全定义的，则从 A 端口退出；否则从 0 端口退出。以下伪代码描述了这些函数：

```
1 updateA(_, traj::Traj) = (homomorphisms(B,last(traj)), length(traj)) 2 updateC ((
bnext:bs,i), traj::Traj) =3 ([b for b in bs if !isnothing(postcompose(traj ,b,i))] ,i) 4 updateC
(([] ,i::Int),_) = error 5 6 readout (( bnext:bs,_), traj::Traj) =return (B,traj +bnext) 7
readout (([] ,i::Int), traj::Traj) =case postcompose(traj ,traj [i] ,i) of 8 nothing => return
(0, traj + initial(last(traj))) 9 new_a_x => return (A, traj + new_a_x)
```

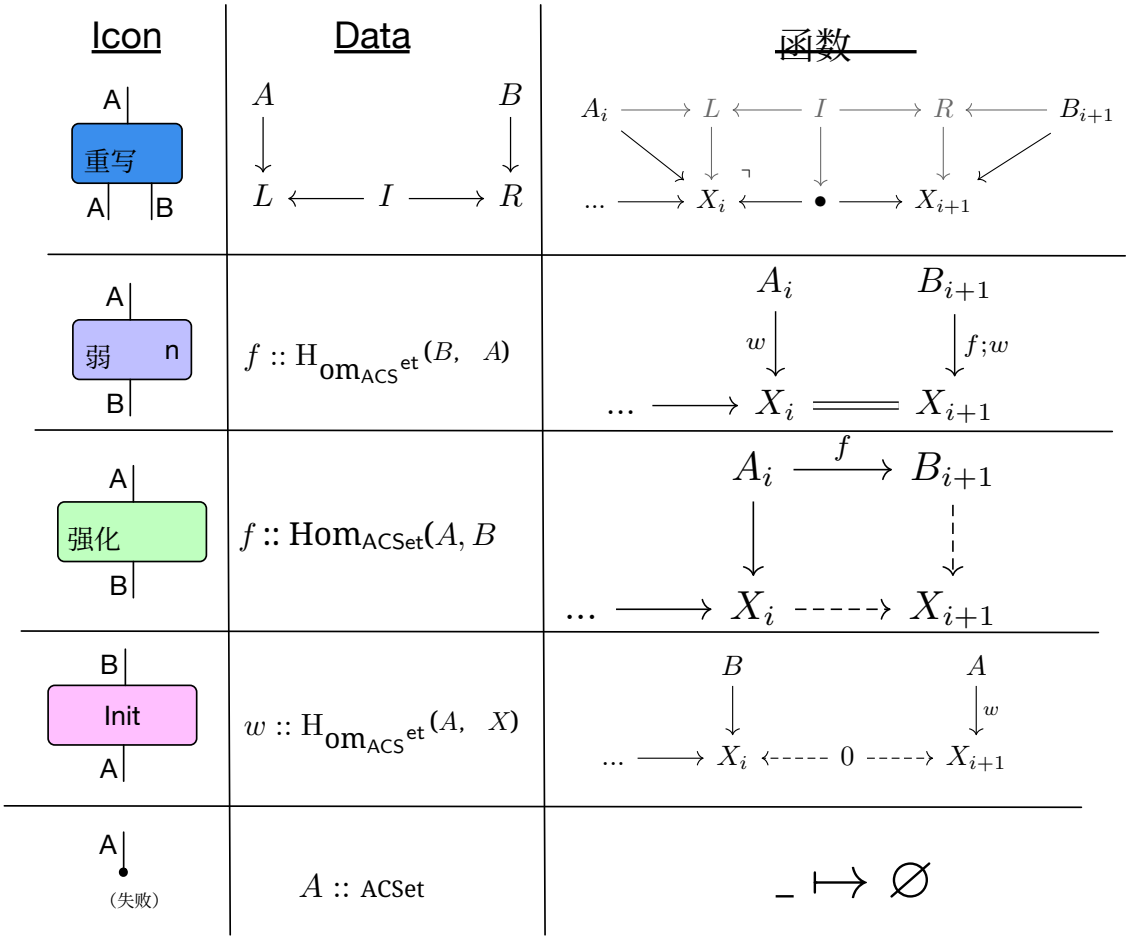


图 8：具有平凡动力学的图重写程序的原始生成态射，即它们是从输入端口到输出端口的纯函数，失败除外，它可以引发异常。

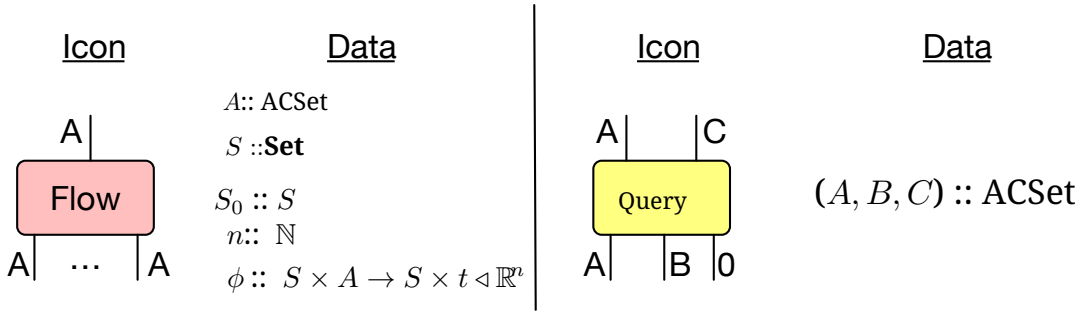


图 9：用于具有非平凡动力学的图重写程序的原始生成态射。**左侧**。纯控制流的生成器。分配给每条出线的值 $\mathbb{R}^{\geq 0}$ 是一个权重，用于调整通过该端口退出的概率。**右侧**。对当前世界中的每个主体 B 运行子程序 $B \to C$ 的生成器， X_i。

尽管使用 `Query` 盒的“标准”方式是将一个子程序从 B 输出端口连接到C 输入端口，从而形成一个 $A \to A + 0$ 接口，但它也可以用更灵活的方式使用。例如，一个将规则 rw 应用于世界状态中满足属性 ϕ 的某个（任意） A 的过程，如图 10 所示。

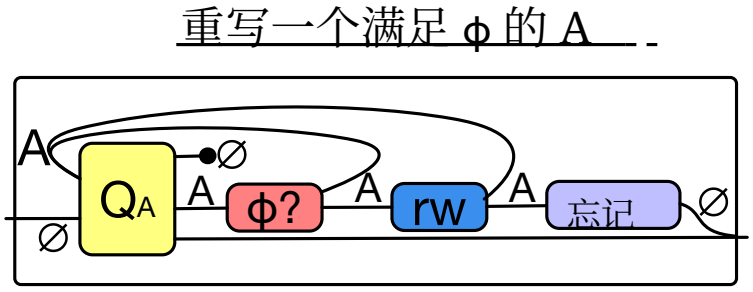


图 10：Maybe 单子上下文中的程序 $0 \to 0$。其输入是一条轨迹，当前时间步中包含一个空主体。然后，我们遍历所有形状为 A 的可能主体。一旦找到一个既满足性质 ϕ 又成功完成重写的主体，就用唯一的 0 形主体替换我们的焦点为 A 形的主体并退出。

上述原语旨在兼具易于理解的特点（例如，控制流、焦点转移和状态变更各自独立）以及足够的表达能力，以复现流行的基于主体的模型。然而，由于规范要求新原语必须包含 $\mathbf{DK}_t$ 中态射的数据，因此这些原语可以通过有章可循而非临时拼凑的方式进行扩展。此外，这些原语可以组合成更高抽象层次的操作库，供领域专家使用。

4.3 实现

我们的实现对于定义 2 中的有限例外多项式单子是通用的，并使用余数据结构来表示（可能无限的）行为树。这些树可以直接为每个生成器提供，不过从米利机表示以编程方式生成它们更为方便。将 $t =$ Maybe 切换为 List + 1 or Dist + 1，即可得到非确定性或概率模拟。对于许多图重写示例，我们希望考虑所有可能的匹配，而不只是任意一个匹配。实现这一点的规范方法是使用一个重写，它会生成一个输出列表，其中包含所有可能匹配所对应的输出。如果要构建程序，以通过一组重写规则进行实证测试，从而确定哪些图可以到达，那么这一点尤为重要。此外，如果某些匹配比其他匹配更有可能出现，那么可以将定义在此输出列表上的分布纳入程序中。

利用我们的重写程序形式体系的一种方式是函子数据迁移 [24]。给定一个 ACSet 模式之间的函子 $S \to T$，一种 Σ 迁移会以普适的方式将 S ACSet 推前为 T ACSet，而 Δ 迁移则会朝相反方向迁移数据。由于这是函子性的，因此可以将这些迁移应用于 ACSet 态射、重写规则以及整个图重写程序。以图形方式工作的另一项关键优势在于连线图的各种组合形式，它们可以简洁地组织带有 $\otimes$ 和 $\fatsemi$ 的态射，也能通过运算子代入以层级方式构建程序。

4.4 示例：离散 Lotka–Volterra 模型

NetLogo 网站 [26] 上展示的第一个基于智能体的模型，是一个狼羊捕食模型。图 11 展示了使用当前框架构建的这一模型的类似实现。其构建运用了 Σ 和 Δ 数据迁移、运算子代入、 t = Maybe，以及原语 重写、削弱、`ControlFlow`和`Query`。随附笔记本提供了该模型构建过程及运行方法的教学式讲解。随附笔记本中还提供了一个使用 Dist 单子在$\mathbf{DK}_t$ 中构建模型的示例。

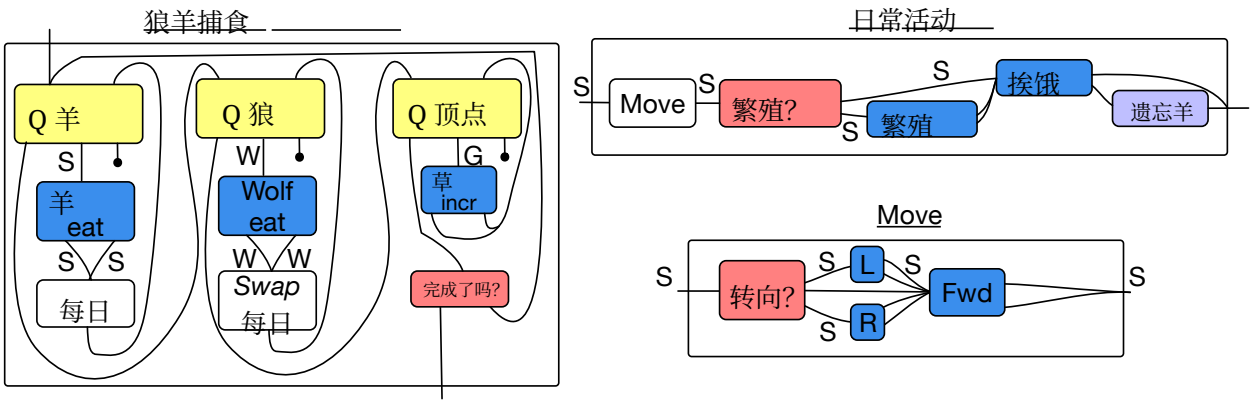


图 11：执行狼羊模拟的程序。此处的交换表示一种 Δ 迁移，它在图 6 的模式中交换狼和羊，从而使狼和羊共有行为的实现得以共享。智能体形状 0 用一条无标签线表示；不过，共享同一目标端口的线不必全部带有标签，因为它们必须共享同一个主体，程序才能具有良好的类型。

5 结论与未来工作

动力学追踪的一般理论指导了图重写领域专用语言的开发。由于这种语言以数学方式理解，并表示为组合数据，而非编程语法，因此高级重写过程可以被理解和序列化，且独立于任何特定实现（尽管我们开发了一个 Julia 实现）。由于这些过程被解释为$\mathbf{DK}_t$中的态射，数据迁移和各种形式的组合等抽象操作也能够自然地应用于这些过程。此外，通过多项式函子融入米利机、系统接口和行为以及单子效应，启发我们实现了一种简洁且通用的方案。

这项工作可以沿多个方向扩展。我们提出的形式体系聚焦于这样一种动力系统：它由单个主体与之交互。在主体与其他系统交互时，该系统保持冻结；但在考虑多个主体同时参与的情形时，这一假设可以放宽，也就是说，可以实现结合图变换的并行编程。这里提出的轨迹概念要求所有涉及的 ACSets 共享同一模式（因为不同模式的 ACSets 之间不存在态射）；这一概念也可以推广，以支持多尺度建模（既在高层的“宏观”模式中进行数据变换，也在低层的“微观”模式中进行数据变换）。

我们计划用这种形式体系表示更多现有的基于智能体的模型，并开发新的模型。在存在涌现效应的各种情形中，基于智能体的模型都是首选的建模方式，例如

物理现象（例如流和扩散模拟） [3], 人类交通网络 [22], 以及流行病学 [10]。为 Catlab 增加增量图匹配支持 [9] ，对于让重写程序在性能上能够与 NetLogo 和 Kappa 等成熟软件竞争至关重要。

参考文献

[1] Ahman, D., Uustalu, T.: 有向容器作为范畴。Electronic Proceedings in Theoretical Co mputer Science **207**, 89–98（2016年4月）。doi:10.4204/eptcs.207.5, https://doi.org/10.4204% 2Feptcs. 207.5[2] Blostein, D., Fahmy, H., Grbavec, A.: 图重写在实际应用中的问题。载于：图文法及其在计算机科学中的应用：第五届国际研讨会，威廉斯堡，美国弗吉尼亚州，1994年11月13–18日，论文选集 5。第38–55页。Springer（1996）[3] Bonabeau, E.: 基于主体的建模：模拟人类系统的方法与技术。美国国家科学院院刊 **99**（增刊_3），7280–7287（2002）[4] Brown, K., Patterson, E., Fairbanks, J.P.: C-集合的双推出重写。CoRR **abs/2111.03784**（2021），https://arxiv.org/abs/2111.03784[5] Bunke, H.: 属性化程序图文法及其在示意图解释中的应用。IEEE模式分析与机器智能汇刊（6），574–582（1982）[6] Corradini, A., Heindel, T., Hermann, F., König, B.: 半推出重写。载于：国际图变换会议。第30–45页。Springer（2006）[7] Ehrig, H., Pfender, M., Schneider, H.J.: 图文法：一种代数表示方法。载于：第14届开关与自动机理论年度研讨会（SWAT 1973）。第167–180页。IEEE（1973）[8] Fahmy, H.: 通过图重写处理不确定性条件下的推理。（1996）[9] Fan, W., Wang, X., Wu, Y.: 增量式图模式匹配。ACM数据库系统汇刊（TODS）**38**（3），1–47（2013）[10] Hunter, E., Mac Namee, B., Kelleher, J.D.: 人类传染病流行病学中基于智能体模型的分类体系。人工社会与社会模拟期刊 **20**（3）（2017）[11] Joyal, A., Street, R., Verity, D.: 带迹幺半范畴。载于：剑桥哲学学会数学学报。第119卷，第447–468页。剑桥大学出版社（1996）[12] Kelly, M.: 富化范畴论基本概念，第64卷。CUP Archive（1982）[13] Löwe, M.: 单推出图变换的代数表示方法。理论计算机科学**109**（1-2），181–224（1993）[14] Niu, N., Spivak, D.I.: 多项式函子：交互的一般理论[15] Overbeek, R., Endrullis, J., Rosset, A.: 使用 pbpo 进行图重写与重新标记+。载于：图变换：第14届国际会议，ICGT 2021，作为 STAF 2021 的一部分举行，线上活动，2021年6月24–25日，会议论文集。第60–80页。Springer（2021）[16]Patterson, E., other contributors: Algebraicjulia/catlab.jl：v0.13.5（2021年12月）。doi:10.5281/zenodo.5771194, https://doi.org/10.5281/zenodo.5771194[17]Patterson, E., Lynch, O., Fairbanks, J.: 面向技术计算的范畴数据结构。Compositionality **4** （2022年12月）。doi:10.32408/compositionality-4-5, https://doi.org/10.32408/ compositionality-4-5[18] Patterson, E., Spivak, D.I., Vagner, D.: 连线图作为对称幺半范畴中计算的范式。Electronic Proceedings in Theoretical Computer Science **333**, 49–64（2021年2月）。doi:10.4204/eptcs.333.4, https://doi.org/10.4204%2Feptcs.333.4[19] Plump, D.: gp 2 的设计。arXiv预印本 arXiv:1204.5541（2012）[20] Schürr, A.: Progress：一种基于图文法的 vhl语言。载于：图文法及其在计算机科学中的应用：第四届国际研讨会，德国不来梅，1990年3月5–9日，会议论文集 4。第641–659页。Springer（1991）[21] Selinger, P.: 幺半范畴图形语言综述。载于：物理学新结构，第289–355页。Springer Berlin Heidelberg（2010）。doi:10.1007/978-3-642-12821-9_4, https://doi.org/ 10.1007%2F978-3-642-12821-9_4

[22] Smith, L., Beckman, R., Baggerly, K.: Transims：交通分析与仿真系统。技术报告，洛斯阿拉莫斯国家实验室（LANL），洛斯阿拉莫斯，新墨西哥州（美国）（1995）[23] （https://math.stackexchange.com/users/666875/richard southwell），R.S.: 富化范畴中的余积是什么？Mathematics Stack Exchange，https://math.stackexchange.com/q/4353966，URL:https://math.stackexchange.com/q/4353966（版本：2022-01-11）[24] Spivak, D.I.: 函子数据迁移。Information and Computation **217**，31–51（2012）[25] Spivak, D.I.: Poly 上范畴结构的参考资料 **Poly** （2022）。doi:10.48550/ARXIV.2202.00534, https://arxiv.org/abs/2202.00534[26] Tisue, S., Wilensky, U.: NetLogo：一种用于复杂性建模的简单环境。载于：国际复杂系统会议。第21卷，第16–21页。CiteSeer（2004）[27] Voss, C., Petzold, F., Rudolph, S.: 工程设计中的图变换：过去十年的概述。AI EDAM **37**，e5（2023年）[28] Wisnesky, R.: 最小化单子推导式。技术报告，CiteSeer（2011）